Edinburgh Napier
UNIVERSITY

THE SERVERSIDE

Web Tech
SET08101

Simon Wells
s.wells@napier.ac.uk
http://www.simonwells.org

TL/DR

Edinburgh Napier UNIVERSITY

- Developing **client** sites locally is all well & good, but it doesn't help us to share our work with others.
- Or to do more **dynamic** & **social** stuff
- To do that we need **servers**

Edinburgh Napier UNIVERSITY

AIMS

- At the end of this (sub-section) of the topic you will be able to:
  - Static versus Dynamic serving
  - Can have lots of functionality on the server side
  - Node.JS is one way to get that functionality

Edinburgh Napier
UNIVERSITY

CLIENT SIDE DEV

- So far mostly client-side development - most HTML, CSS, & JS will run quite happily in the browser; served up from the local file system - what are the drawbacks?

- Only local users (not web users)

- No storage (NB. Browser storage API) - restrictions on writing to local file-system from browser

- Can use a web server to host our sites - make them available to web users - still fairly static - interaction is only what JS enables - mostly structured around the HTML document

Edinburgh Napier UNIVERSITY

STATIC WEB SERVING

- Nothing wrong with this - vast majority of sites are wholly static:
  - HTML + CSS + JS (presentational)
- Some are quasi static (our cipher sites are a good example):
  - HTML + CSS + JS (handles user interaction, performs calculation)
- (Quasi-) Static serving is **very** scalable & performant
  - Geo/Neo-cities - hosted thousands of static sites across two reasonably sized web servers
  - GitHub pages (static serving) - again, used to host thousands of sites for programming projects/ documentation on a handful of reasonably sized web servers (rest of GitHub needs way more servers)
- Take care of resources (e.g. size of images & other media, minify JS,) to minimise bandwidth
- All you're really doing is serving up text (HTML, CSS, JS are all text) - this can be optimised for performance

Edinburgh Napier UNIVERSITY

DYNAMIC WEB SERVING

- As soon as the server has to do anything more than send back data (HTML,CSS, JS) on receipt of a request - performance is impacted
  - NB. At what point does a web-site become a web-app?
- The more server-side calculation - the more impact, i.e. database access, manipulating data, generating web-pages
- However server-side dev can be exciting - without server-side development we wouldn't have social media, forums (where would you be without stack overflow?), comments, real-time comms (slack?), games, &c.

Edinburgh Napier
UNIVERSITY

WHAT KINDS OF DYNAMISM?

- Generate dynamic page content
- Create, open, read, write, delete, close files on the server
- Collect form data from web pages
- Create, Read, Update, Delete (CRUD) data in data stores

Edinburgh Napier
UNIVERSITY

WHAT HAPPENS?

- Still the Web (HTTP+HTML/CSS/JS) but instead of just listening for a request,
  - i.e. GET index.html then returning the requested resource (index.html)
- We execute some function/program on the server (depending on the request), calculate a result, generate a response (HTML but also JSON, XML, & others), then send the response back.
- This is all more complex (resource intensive) than just reading data from disk and sending it back.

Edinburgh Napier UNIVERSITY

SERVER SIDE DEV LANGUAGES

- JavaScript - we've already played with this for client side apps; why not use it to add functionality on the server as well?
- Python - (used in adv. web tech.) A popular general purpose language - very widespread in scientific/data science areas
- PHP - Still widespread but falling out of favour (persistent security & performance issues)
- ASP - A Microsoft technology. Used to be widespread amongst Microsoft-based organisations
- Java - Client-side (applets) & Server-side (servlets)
- PERL - Used to be the standard ('90s\'00's). A lot less widespread now.
- CGI (Common Gateway Interface) - A standard for Web servers to interface with other programming languages, e.g. C, C++, Perl,

Edinburgh Napier
UNIVERSITY

JS ON THE SERVER

- Node.JS is the standard
- Wasn't the first (Netscape Livewire Pro) but definitely won the popularity competition
- Hugely popular over last few years - greatly influenced reliability of deployment, availability of support & of libs
- "JavaScript everywhere" - advantages to a unified programming language approach for the web
- For our purposes: Work with JS in client & server rather than learning yet another language

Edinburgh Napier UNIVERSITY

NODE.JS: OVERVIEW

- An open-source, cross-platform, run-time environment for JS
- Primarily used for web-servers & networking related JS code:
  - i.e. JS is executed on the server & the results packaged up & returned to client
- Event-driven architecture with asynchronous IO
  - Aims to optimises throughput & scalability for web-apps
- Designed to address limitations (at the time) of dominant server software (Apache) to handle lots of concurrent connections (>10,000) without blocking

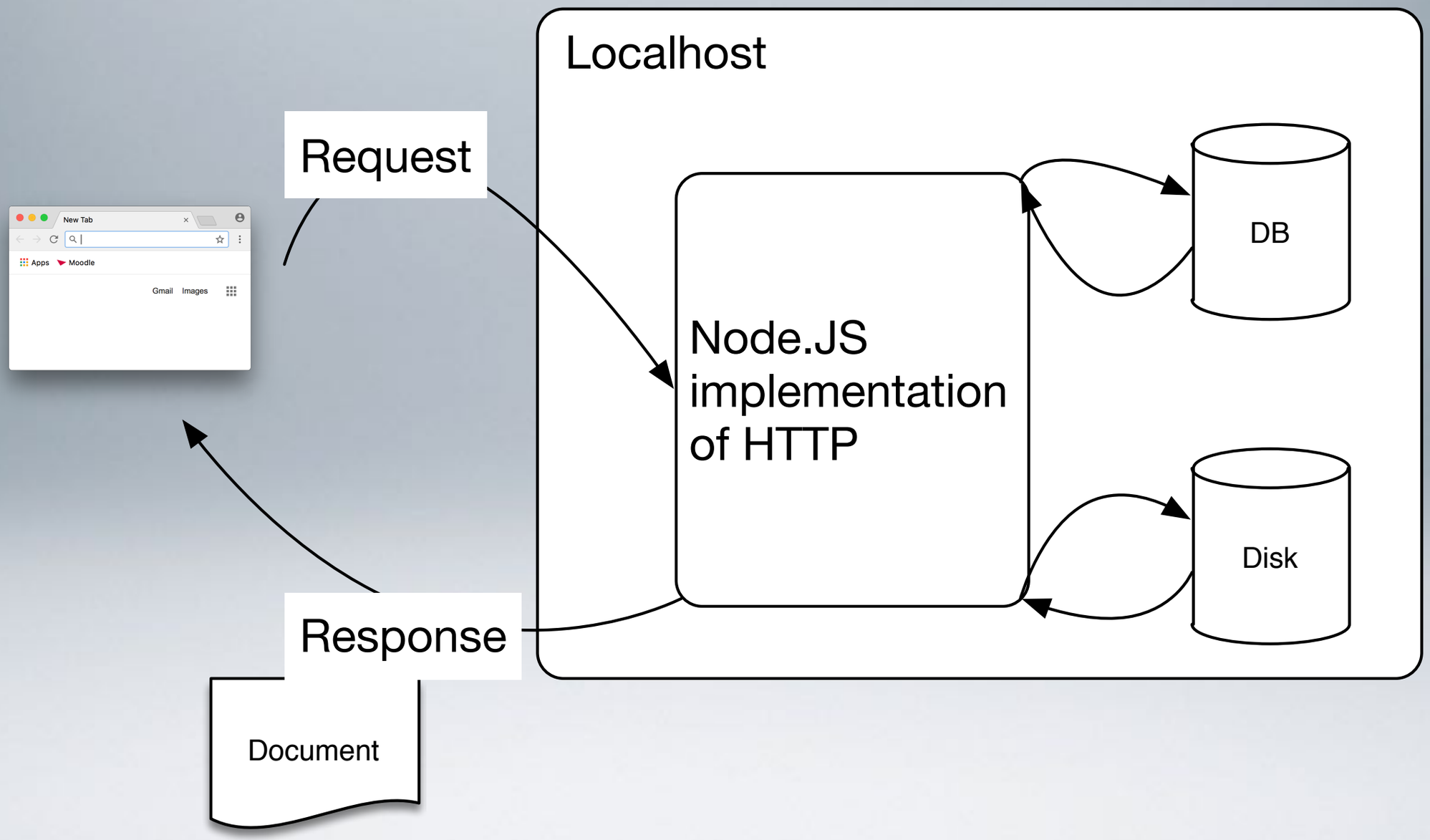
NODE.JS: PLATFORM

- Google V8 JS engine - fast, stable, capable
- Event driven: Write functions to handle events, when event occurs the function is called
- Non-blocking/Asynchronous: Doesn't wait for function to complete before moving on to the next thing - can be a limiting factor on other platforms,
  - e.g. reading a file from file system: tells the OS to do it then processes next request. OS tells node that it is finished (callback) & sends read file - no waiting for potentially long tasks to finish

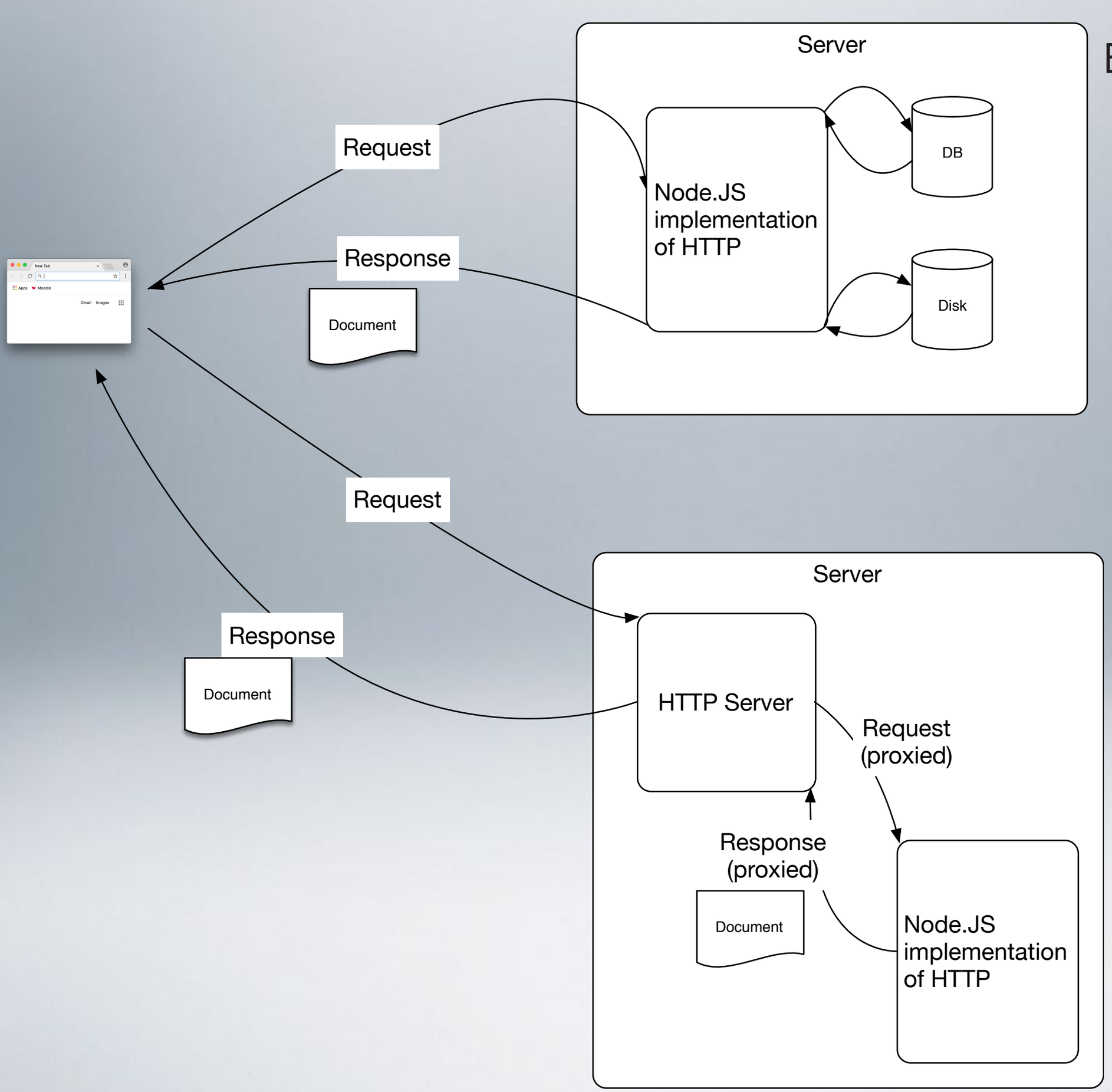
Edinburgh Napier UNIVERSITY

NPM

- Node Package Manager (NPM)
  - A way to package up & distribute your code
  - A way to take advantage of/build upon libraries of existing code
- Install, update, uninstall libraries of code called Modules
- Modules for File I/O, Networking (DNS, HTTP, TCP, UDP, SSL), binary data (buffers), cryptography, data streaming, & many more


Localhost
Request
New Tab
Apps
Moodle
Gmail
Images
Node.JS
implementation
of HTTP
DB
Disk
Response
Document

Edinburgh Napier UNIVERSITY



Edinburgh Napier UNIVERSITY

SUMMARY

- Static versus Dynamic serving
- Can have lots of functionality on the server side
- Node.JS is one way to get that functionality

Edinburgh Napier
UNIVERSITY

NEXT

- More Node.JS & RESTful Design